

## SourceGuardian™ Loaders

Scripts protected with SourceGuardian™ will require the installation of a SourceGuardian™ loader on the target machine in order to run. Protected scripts will automatically attempt to find the loader in the `ixed/` directory located within the protected script's directory or parent directories. SourceGuardian™ loaders may also be installed into a `php.ini` configuration file - This is useful, for example, if automatic loading is not supported or if faster performance is required.

For PHP versions 5.2.5+, SourceGuardian™ loaders need to be installed in the PHP extensions directory (`extension_dir`). You may find the `extension_dir` path in the `php.ini` configuration file or in the `phpinfo()` output.

The way the dynamic loading `dl()` function works in PHP has been changed since version 5.2.5 - It may load PHP extensions located **ONLY** in the `extension_dir` directory or a subdirectory within it. This means that SourceGuardian™ loaders cannot be loaded automatically from the `ixed/` directory located within the protected script's directory or parent directories for PHP 5.2.5+. Usually you will get the following error message in that case: "Warning: `dl()` [function.dl]: Temporary module name should contain only filename". Please also read the note below about installing the loader for PHP 5.2.5+.

## Automatic loading

Many operating systems and installations of PHP will load the Protected scripts without any modification. PHP will be able to find and load the appropriate loader if the following conditions are met:

1) Operating system and PHP mode:

Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as CGI or CLI

Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as a webserver's module (with thread safety off)

Windows - PHP installed as CGI or CLI

2) Thread Safety is disabled. You may check `phpinfo()` output for this.

3) `dl()` is enabled. You should have `enable_dl=On` in your `php.ini`.

4) The PHP extensions directory (`extension_dir`) needs to exist. Please check that the `extension_dir=` option in `php.ini` points to the real directory. Some hosting companies have incorrect installations of PHP and this can cause problems.

5) The latest `ixed` loaders are installed in an `ixed/` subdirectory within your scripts directory or any parent directory.

6) PHP version is below 5.2.5.

Please note: if your server and PHP configuration conform to all conditions above for automatic loading except only a PHP version, then it is enough to copy an appropriate loader into the PHP extension directory (`extension_dir`). The loader will be used automatically from the `extension_dir` directory - no need for changes in the `php.ini` configuration file.

## Manual installation

It is possible to manually install the loader and this is required in the following conditions:

1) Operating system and PHP mode:

Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as a webserver's module (with thread safety on)

Windows - PHP installed as Apache module (thread safety is always on)

2) If Thread Safety is enabled. You may check `phpinfo()` output for this. PHP installed as a webserver's module under Windows will always have Thread Safety on.

3) If `dl()` is disabled. You have an "`enable_dl=Off`" setting in the `php.ini` configuration file.

Manual installation requires permissions to access the `extension_dir` directory and the `php.ini` configuration file. Manual installation may be used even if automatic loading is available. With appropriate and manually

installed SourceGuardian™ loaders you give the maximum performance for your protected scripts. This

is because the script will not need to search for a loader each time it runs.

To install the SourceGuardian™ loader manually you need to do the following:

- 1) Choose an appropriate loader for your operating system and version of PHP. Please refer to the "Loader filename structure" section below to know which loader is required for your operating system and version of PHP.
- 2) Find the loader file in the ixed/ subdirectory within SourceGuardian main installation directory and copy it to the PHP extension directory (extension\_dir - check the phpinfo() output).
- 3) Find the location of the php.ini configuration file (check the phpinfo() output) and add "extension=ixed.X.X.YYY" directive at the end of the file. (X.X is the major version of PHP and YYY is the name of operating system)
- 4) Restart the webserver in order to apply changes done in the php.ini configuration file and reload PHP.
- 5) Optionally you may check the phpinfo() output now to find out that SourceGuardian loader was successfully installed - search for "SourceGuardian" within the output.

## Loader filename structure

The following provides an overview of the loader naming conventions:

ixed.X.Y.Zdd.os

X.Y - major PHP version number (4.3 for 4.3.x, 5.0 for 5.0.x, 5.1 for 5.1.x, 5.2 for 5.2.x)

Z - minor PHP version number (2 for 5.0.2) This is optional and most loaders DOES NOT have the minor version number in the file name.

This part may be missed in the loader name which means that this loader is for all higher PHP versions:

ixed.4.3.lin - for all PHP 4.3.x versions

ixed.5.0.0.lin - for PHP 5.0.0 only

ixed.5.0.1.lin - for PHP 5.0.1 only

ixed.5.0.2.lin - for PHP 5.0.2 only

ixed.5.0.lin - for all PHP 5.0.3+ versions and higher

ixed.5.1.lin - for all PHP 5.1 versions and higher

ixed.5.2.lin - for all PHP 5.2 versions and higher

dd - optional code of supported encoder and system:

(missed) - this loader will load scripts protected with full version of SourceGuardian™

ts. - this loader will load scripts protected with the full version of SourceGuardian™ and is suited for manual installation on systems with Thread Safety enabled PHP (see details above)

os - three char code of operating system type. Currently supported:

.win - Windows

.lin - Linux

.fre - FreeBSD

.net - NetBSD

.ope - OpenBSD

.sun - SunOS

.dar - MacOSX (darwin)

.hp- - HPUX PA-RISC system

For some operating systems there are different versions of loaders for 32-bit and 64-bit mode. File names of such loaders are the same as it is impossible to determine 32/64-bit mode on a PHP level. Although 32-bit and 64-bit loaders are packed in different zip (tar.gz, tar.bz2) files so you can easily determine them. You need to use the correct 32-bit or 64-bit version of the loader on your system according to the platform and the mode PHP executable or shared object is built. You **may safely try** 32-bit version and then 64-bit one if you are unsure. Usually you will get the following error message in the case of wrong

32/64-bit loader is installed: "Unable to load dynamic library 'ixed....' cannot open shared object file" or "Unable to load dynamic library 'ixed...' wrong ELF class: ELFCLASS32(64)". If you have access to a command line shell you may check the PHP using command line "file" tool, e.g. "file /path/to/php".